Cyberattack Prediction Through Public Text Analysis and Mini-Theories

Ian Perera, Jena Hwang, Kevin Bayas, Bonnie Dorr, Yorick Wilks
Florida Institute for Human and Machine Cognition
Ocala, Florida, USA
{iperera, jhwang, kbayas, bdorr, ywilks}@ihmc.us

***Abstract*—This paper describes a new approach to detection and tracking of potential cyberattacks from analyzing large quantities of cyber-related webpage text, using ontological knowledge about such attacks combined with composable causal models represented in Probabilistic Soft Logic. The *stages* of a cyberattack kill chain are viewed as a sequence of both observed and unobserved events (e.g., *reconnaissance, weaponize, exploit, install*) and explicit mentions of, or related to, such events are examined as potential signals for a future attack. Using a suite of natural language processing techniques, sentences from input news texts are automatically classified according to the described cyberattack event, then enriched with named entity recognition for the rapid detection of key elements that might be associated with potential cyberattacks. We present our work as a framework for rapid and flexible predictive analysis of the ever-increasing amount of cyber-related text data, with initial experiments indicating that event detection using parsing and named entity recognition combined with statistical relational learning show promise in time-series prediction from news text.**

***Index Terms*—Cybersecurity, Statistical Relational Learning, Natural Language Processing, Big Data**

I. Introduction

While target- and attack-specific data play a vital role in cybersecurity, a currently underutilized source of information for cyberattack prediction and preparation is text found in big data on the web. News articles and reports provide sociopolitical context for cyberattacks, revealing possible threat actors, motivations, and methods. The expanding number of reports of cyberattacks can make deeper analysis of such data prohibitively time-consuming. However, shallow text analysis cannot provide much of the detail necessary to yield actionable steps for improving security measures against the vast number of cyberattacks occurring each day. For example, reports of a distributed denial-of-service (DDoS) attack may be predictive of future attacks if the relevant organization is related to the reported targets, or a security breach of an internet-facing service may predict a future attack on companies using the same service.

This research is supported, in part by the Institute for Human and Machine Cognition, and in part by the Office of the Director of National Intelligence (ODNI) and the Intelligence Advanced Research Projects Activity (IARPA) via the Air Force Research Laboratory (AFRL) contract number FA875016C0114. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government.

Cyberattacks and malware proliferation are often only the final event in a sequence of prior events, such as exploitation of a bug in software, identity theft, or scanning for vulnerabilities. Understanding these relationships between such events and what kinds of signals can be gleaned from news reports and webpages is key to predicting future cyberattack events. Thus, in contrast to prior work that attempts to simply find mentions of attacks in Twitter [1], [2], we view detected events as visible indicators of a sequence of cyber events (e.g. reconnaissance, delivering exploits, compromising systems, etc.) occurring over the course of a cyberattack objective, which together form the "cyberattack kill chain" [3].

II. Natural Language Processing and Analysis of Cyberattack Newstext

To enable deep analysis of these events tailored to a particular organization, or to analyze a certain type of cyberattack, we have developed a multi-step processing system with automatic text analysis for extraction of events and their participants combined with a method for testing and training human-readable, rule-based hypotheses (mini-theories) about extracted events, entities, and possibly other related data. However, deep analysis of text data typically requires computation time that is infeasible for the large volume of cyberattack news data we process (thousands of documents per day), and is often not robust to the variety of text data formats we encounter on the web (e.g., headlines, tweets, etc.). To address this issue, we develop a "flexible parsing" method, quickly extracting meaningful data from large amounts of text without relying on grammatical structure.

In natural language processing (NLP), event detection involves finding information about entities in events, the role they play, and times and locations of the event. For example, in the sentence, "Kennedy was **shot dead** by Oswald," the event type is *Conflict.Attack*, the entity *Kennedy* plays the role of a *Target* and entity *Oswald* plays the role of an *Attacker* in the event. Event detection has been applied as a sub-component of many different applications, including the forecasting of future or related events in the domains of protest tracking [4] and stock market prediction [5].

Figure 1 shows an example of a text with events belonging to various stages of a cyberattack in red that might serve to predict future attacks, generated by one configuration of our system described in Section IV-A. In this example, the

detection of stated vulnerabilities in software signals the potential for future cyberattacks (e.g. a hacking event), which can in turn generate warning signals for the victimized party, opening up the possibility of preventing future attacks.

The Trump Organization's mail servers all run on an unpatched version of **[Microsoft]-VENDOR** Windows Server 2003 […], making the servers extremely vulnerable to **[hacking]-AOO:general** and cyber **[attack]-AOO:general**, security researcher Kevin Beaumont **[discovered]-DETECT**. Moreover, Trump's webmail runs on **[Microsoft]-VENDOR** **[Exchange]-APP** 2007 (SP3 RU16), a version with a high number of well-documented **[vulnerabilities]-AOO:data**.

Fig. 1. An example of a news report with our system's output of events (red) with ontological labels (black) at different stages of the cyberattack kill chain, detailing vulnerabilities that may be predictive of future cyberattacks. Named entities (blue) relevant to cyberattacks are also presented.

The detection and tracking of developing events is a fundamental aspect of both text understanding in general and in high volume data analysis scenarios. Such a capability is seen as a strategic aspect of understanding: the most significant and potentially dangerous items mentioned in a text are in the form of events, most notably events that are potential cyberattacks in this case.

The events comprising cyberattack stages are defined by Lockheed Martin's cyber kill chain [3]. An extended version of this chain incorporating sub-event details and stages preceding the attack is shown in Figure 2. The stages of a cyberattack are viewed as potentially observed events (e.g., *reconnaissance, weaponize, exploit, install*), and explicit mentions of, or others related to, such events are examined as potential signals for a future attack. Figure 3 illustrates the set of events and sub-events (described in detail in Section III) under examination as indicators of potential future attacks. While there currently exist labeled corpora for attack descriptions in cybersecurity reports [6], we are unaware of existing corpora for such reports of attacks appearing in the news, which combine a higher level of analysis of the attacks with possible attackers, victims, and ramifications.

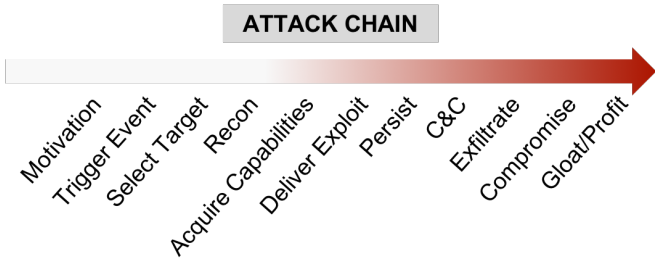

Fig. 2. Events in the Lockheed Martin Cyberattack Kill Chain with pre- and post-attack events added to reflect events potentially mentioned in web text.

The assumption behind the experiments described in this paper is that rich knowledge structures in the form of small causal models, i.e, mini-theories [7], provide a more appropriate and tractable methodology for this purpose than methods drawn from philosophically motivated action-event

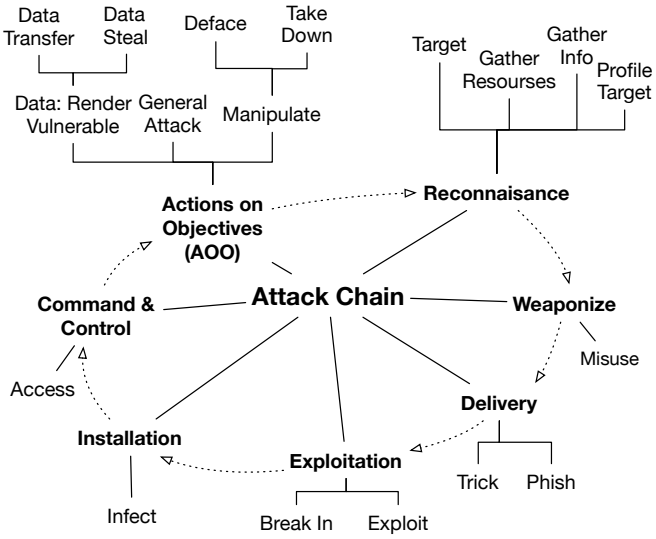

Fig. 3. Events in the Cyberattack Kill Chain in connection with the Threat Ontology, with connections between events and sub-events

theories. Rather than trying to capture all possible details in a Davidsonian manner [8] and put them into a general purpose reasoner to generate inferred events, we use mini-theories to provide a background of how various entities and events relate to each other in the cybersecurity domain, specifically within the attack chain model. This approach reduces the search space for possible inferences, making inference faster while also allowing for inference from incomplete data when methods of entity and event detection fail.

A key component of our approach is the capability for "flexible parsing" that allows event extraction to look beyond syntax to determine semantic relevance without requiring an exact determination of grammatical structure, providing both the speed to handle large amounts of data and the flexibility to handle a wider variety of data sources with varying degrees of grammaticality. Given the semantic information provided by natural language processing, our key predictive leverage then comes from the encoding of relevant knowledge in mini-theories, a well-tested and implemented theoretical structure with some years of development behind it. Initial experiments indicate that these system components and analysis methods using statistical relational learning show promise as a tool for predictive modeling given event extraction as a reasonably strong signal.

There has been a great deal of NLP work on event detection over many years, ranging from syntactic patterns in the early work of Riloff [9] to their combination with statistical classifiers [10]. In this work, we compare two methods for extracting events as signals to a future impending attack—lexical items inferred from mentions (or "trigger words") with the term frequency providing the signal for detection of events through the aggregation of many detections over time, and sentence-level word embedding classification for detecting sentences describing specific events. Both capture the notion

that widespread reporting of an event provides a strong signal that a significant or actionable event has occurred.

These signals then serve as data for the statistical relational learning models, represented as rules using Probabilistic Soft Logic (PSL) [11]. These rules consist of human-generated relations (in this case, created by the authors to test the system) to represent automatic analysis of semantic data, such as specific events or entities with possible additional parameters. We then perform weight learning and inference over the hand-built mini-theories, populated with event detection values to make predictions about future attacks, similar to earlier work on deriving plan and intention processes using Cascading Hidden Markov Models [12].

The extraction of concepts from unrestricted text has been mentioned in prior work, e.g., by Jones [13], but these have focused on extracting concepts from databases. Our work focuses on event detection and threat tracking from unrestricted text found in webpages, which includes a wide variety of text modalities, from headlines, to tweets, to news reports, each with differing syntactic structure. To perform analysis on this text and generate structured representations of events and their associated entities, we first process and classify sentences from scraped news text articles containing the words "cyber" and "tech" linked to by tweets to extract events (yielding 800-3000 documents per day). We then perform named entity recognition to detect mentions of potentially relevant participants of the event, and finally develop relational models to make predictions by generating a time series of expected future attacks.

III. Threat Ontology Development

An ontology provides the framework for event detection, enabling interpretation of the detected events, what participants we might expect in the event, and how events relate to one another (e.g. causality, necessity, subsumption etc.). We pursued two distinct goals in developing the ontology for this work. First, the ontology types should be fine-grained enough to pursue deep analysis of the various stages of a cyberattack, with at least some of the stages being potentially observable through reports of publicly available data. Secondly, the events should be distinct enough that a human judge can distinguish between descriptions of the different events in text. This distinction then allows us to develop text analysis tools that provide signals with a minimum of semantic overlap.

Semantic labels used for distinguishing between classes of events are taken from a threat ontology designed and developed to capture event types relevant to cyberattacks (see Figure 3 above). The ontology includes 24 fine-grained semantic types (e.g. general actions on objectives like `ATTACK` for general mentions of attacks, `BREAK_IN` for infiltration events, and `PHISH` sending deceitful, weaponized communication), which are organized into 9 major categories. Seven of these higher level categories coincide with Lockheed Martin's seven steps of the cyber kill chain [3], supplemented with two additional categories including events relating to speech acts and other general but related event types.

The ontology was developed from manual examination of 613 lines of cybersecurity reports from Hackmageddon [14], hand selected by cyberattack experts as containing mentions of a cyberattack belonging to one of the four types: malicious email, malicious URL, attack on internet facing service (ATOIFS), and denial of service (DoS). The text was then annotated and ontologized by a linguistic expert and a trained annotator. All the words providing information relevant to the various stages of a cyberattack were identified, tagged and ontologized to be used in lexical matching. The developed ontology and roles are provided in the Appendix.

IV. Event and Entity Extraction

A. Lexical Matching for Explicit Event Identification

To determine events and entities to be added to our predictive models using the lexical match method, we first enrich the text at a lexical level, capturing both the event mentions and the nominal entity mentions relevant to cyberattack events. The event extraction marks relevant cyber events with semantic labels from the threat ontology (e.g., actions on objectives (AOO) and detection events), and the entity extraction independently marks relevant nominal entities located with a named entity recognition module designed specifically for cyber events (e.g. vendors and application mentions relevant to cyberattack prediction).

With the lexical list gathered from the manually annotated data as the seed set, the lexicon for the nine major categories was expanded using the following lexical resources: TRIPS [15], VerbNet [16], and WordNet [17]. This expanded lexicon was used to develop the basis for the Threat Ontology lexical event extraction. We generate part-of-speech tags, lemmatize input data (removing word endings to achieve consistency in word detection), then perform simple lexical matching to extract explicit event mentions of the threat ontology categories using the expanded lexicon.

While expanded lexical matching can potentially detect more specific events than are indicated by specific terms like "breach" or "vulnerability", it is limited in that such terms are often used in other contexts and therefore there can be a large number of false positives. For example, the term "attack" can be used in multiple contexts, such as medicine, unrelated to cybersecurity. Furthermore, sentences containing these terms may occur multiple times within a document describing the same event yet without relevant information, and the multiple mentions can affect the reliability of the event detection as a signal. In our experiments, this method did not provide usable predictive results, leading us to develop a sentence embedding method.

B. Sentence Embeddings for Explicit Event Identification

Entity extraction remains the same for the sentence embedding method, which we now describe, but rather than using lexical triggers for events, we use FastText [18] to train a sentence classifier to recognize Threat Ontology events at the level of a sentence, rather than at the lexical level. For training data, we used the 2017 Hackmageddon Master

List [14], a record of 950 publicly reported cyberattacks with summary sentences and attack type labels. To prepare this data to train the classifier, we performed an automated mapping of the relatively unstructured attack types in the Hackmageddon Master List to a subset of our Threat Ontology Event types using keyword matching. For example, the phrase "brute force" in the Hackmageddon attack type description would map to ATTACK in the Threat Ontology. In addition to mapping to a subset of events in the Threat Ontology that we believe will be described in news or reports, we create an additional event *GENERIC-CYBER* to handle attack types that do not map directly to the Threat Ontology, but are related to a cyberattack in some way.

To evaluate the accuracy of this method of event detection in sentences, we performed two tests: a binary classifier to identify cyber sentences to predict performance on possibly unrelated text, and a multi-class classifier to identify a subset of the sentences containing Threat Ontology events. For both cases, we trained our sentence classifier on the sentence summaries and their converted attack types in the Hackmageddon 2017 Master List and tested it on the 2018 Master List (updated to February). For the binary classifier, we added a hand-annotated set of sentences containing non-cyber events taken from similar webpage data to provide negative examples.

The binary classifier achieved an AUC of .94, demonstrating that additional filtering of webpage text will probably not greatly improve performance, and that performance improvements would be more likely to come from more training data applied to the classifier itself. Also, the ability of the classifier to identify cyber-related sentences enables us to pull relevant information from sentences occurring among unrelated data, as is often the case in newstext. The multi-class sentence classifier achieved a macro-AUC of .8 for 8 event types. Figure 4 shows the ROC for this classifier. The AUC results for the predominant event types comprising 90% of the sentences in the test data were .82 for *ATTACK*, .91 for *EXPLOITATION*, .84 for *GENERIC-CYBER*, and .94 for *MANIPULATE*.

The sentence embedding method overcomes the limitations of the lexical match method by using the context of the entire sentence to determine which type of event is being mentioned, providing us with reliable data about what event a given sentence is describing. However, the downside of this method is that it requires training data for each class, and distinctions between different types of events are somewhat blurred when taking into account the entire sentence context. Furthermore, the finer-grained distinctions in Hackmageddon reports are not consistently labeled, and do not always correspond with our distinctions. These limitations could be addressed in the future by using training sentences hand-annotated with the Threat Ontology types rather than mapping from the Hackmageddon data.

C. Entity Extraction

Entities are extracted using Spacy 2.0 [19] named entity recognition (NER) system, which is comprised of a residual multi-task CNN trained on OntoNotes 5 combined with a

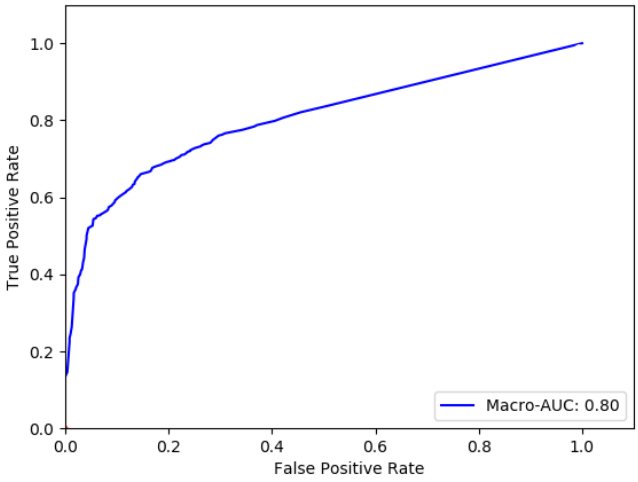

Fig. 4. Macro-ROC curve for the 8-class sentence classifier tested and trained on Hackmageddon data.

corpus of vulnerability databases automatically labeled with software names, vendors, and versions [20]. Entities extracted within a sentence are generally considered to be participants or contextual details (e.g. locations, dates) in that event, but our NER model also includes other relevant terms, such as the method of the attack, the operating system affected, and the language used to exploit a vulnerability. The OntoNotes NER annotation enables detection of geopolitical entities like countries or cities (GPEs), organizations, locations, products, dates, and people, while the NVD-trained [21] model enables detection of relevant cyberattack terms (e.g. *DDoS*, *buffer overflow*), software, hardware, software and hardware vendors, version numbers, files, and affected operating systems.

To preserve performance on the original OntoNotes tags while training for the new cybersecurity tags, we used pseudo-rehearsal [22] by labeling Hackmageddon sentences with the original model (generating only OntoNotes tags) and then presenting them interleaved with the new cybersecurity training instances during the training process. In addition to providing additional event details in a structured format, entity recognition can serve as a signal itself. For example, if an organization suffers a data breach, we would expect to see a large number of mentions of that organization, as well as numerous mentions of related details. Furthermore, we filter event detections containing DATE terms over large timescales (i.e. past years and months) to create a signal that better represents current events rather than reports on historical cyberattacks.

V. A Mini-Theoretic Approach to Event Analysis

Some key features assumed in the detection and tracking of events generally, and in the planned full implementation of this system, are that: (1) events can be characterized as *containers* [23] with sub-events in them, and that are hierarchically related to them, e.g., a container event such as flooding, which could contain a sub-event of a car floating down a street; and (2)

events can be characterized by a changing description that develops in time, so that later sub-events can be related back to the original event under new descriptions.

The assumptions above form the basis of an organization of event definitions around the notion of *mini-theories* associated with domain knowledge. Each mini-theory captures specific knowledge about particular aspects of our world. For example, a mini-theory of data phishing can play the causal role in the mini-theory of software vulnerability. Subsequently, vulnerabilities in software provide the necessary conditions that expedite system exploitation and hacking captured in other mini-theories.

We employ heuristic approaches within a probabilistic framework for deriving mini-theories from clusters of related definitions and testing them on real data using Probabilistic Soft Logic (PSL). PSL is a statistical relational learning framework that allows for weight learning and inference over rules specified in a subset of first order logic. The interface is similar to other statistical relational learning systems like Markov Logic Networks [24], but truth values are considered to be "soft", rather than discrete 0 or 1 values.

This approach enables faster inference and learning, and is also suited to large-scale inference where events can have various confidence levels in their reporting and different levels of detail. While mini-theories are typically presented in a graph representation, they can also be represented using the logical form of PSL—the graph representation simply provides an interface for domain experts, rather than AI researchers or logicians, so as to produce theories and hypotheses that could then be tested in this framework.

To prepare the output of our natural language processing for PSL, we generate predicate files for each event and entity type. Event files will have a value from 0-1 for each day, with this value being either a normalized value with a cutoff based on historical count data, or a proportion of that event's counts over the sum of all types of events detected. We generate one entity file for each entity type (e.g. person, GPE, software, vendor), and also generate a combined file listing the top twenty entities detected that day, with the number of times they were detected normalized to 0-1 using a sigmoid function.

Users can also provide additional files that encode relationships between entities or events. For example, an analyst could define a predicate *OrganizationUsesSoftware* indicating which organizations use a particular type of software, presumably seeing detections of cyberattacks on those organizations as an increased future risk. To define this, the analyst can simply provide a file with each row containing the organization name and the software, along with an optional confidence value. Then a warning can be generated by providing a rule in the following PSL notation:

```
Exploitation(day) & ValidDay(day) &
OrgEntity(organization,day) &
OrganizationUsesSoftware(organization,
software) => PossibleRisk(software)
```

This rule captures an explicit event (mention of a vulnerability (exploit) along with a significant number of mentions of a organization) on any given day and infers that the unnamed software used by the organization may be at risk. The background knowledge of what software the organization uses therefore aids in making predictions from the data, while the logical structure allows the model to make explainable predictions. To generate a more detailed warning, the model can trace the groundings of the variables provided, verify that the weight of the rule is significant enough to be a primary cause of the warning, and then provide additional details regarding the source of the data that triggered the warning.

Note that currently we do not use explicit links between detected events and entities, for several reasons. First, because significant events are likely to be mentioned multiple times, and individual mentions are less likely to yield insight towards meaningful cybersecurity decisions. Secondly, details of the attack may not appear in the same sentence or document that contains the original event mention. However, because the entity search is only run on sentences that describe an extracted event, we can be relatively certain that the extracted entities are involved in some cybersecurity event. In future work, we plan on developing aggregation methods to appropriately weight and combine the numerous individual events with their associated entities at a finer-grained level.

VI. Experiments

A. Event Detection Correlations

Before evaluating the predictive performance of mini-theories using event and entity data, we first evaluate the ability of automatic sentence embedding event detection to detect relevant cyberattack information from unstructured news text. Because our lexical match method led to a very low precision and recall for identifying sentences describing events compared to human judgment, we only evaluate the sentence embedding method for these evaluations, which yielded the promising results shown in Figure 4.

While a correlation with Hackmageddon daily attack counts does not necessarily demonstrate that the same attacks reported through Hackmageddon are being detected, it does show promise in detecting a similar stream of information on the web. Furthermore, while resources such as Hackmageddon rely on human submissions which could be delayed by multiple days, our system can automatically detect news and, therefore, reduce the delay in collecting relevant information.

One limitation of this evaluation is that Hackmageddon counts measure unique cyberattacks, whereas our system will detect multiple reports of the same attack. While the number of detected events can provide a signal as to the severity of a cyberattack based on how widespread the associated reporting is, there is no directly corresponding public ground truth to reflect cyberattack severity in addition to the unique attacks gathered by Hackmageddon.

In an experiment on three months of data, the *GENERIC-CYBER* event was most correlated with Hackmageddon counts, although the *ATTACK* and *EXPLOITATION* events were also significantly correlated. Hackmageddon attack counts are

Data Used	R (Corr)	p-value
Day of the Week	.53	$< .01$
Autocorrelation (prev. day)	.23	$< .05$
Query Return Counts	.55	$< .01$
GENERIC-CYBER only	.48	$< .01$
Events Only	.65	$< .01$
Entities Only	.64	$< .01$
All Events and Entity Counts	**.76**	$< .01$

TABLE I
RESULTS OF MULTIPLE LINEAR REGRESSION TESTS ON HACKMAGEDDON DATA

Data Used	R (Corr)	p-value
Day of the Week + Autocorrelation	.55	$< .01$
All Events and Entity Counts	**.65**	$< .01$

TABLE II
RESULTS OF MULTIPLE LINEAR REGRESSION TESTS ON HACKMAGEDDON DATA FOR ONE-DAY PREDICTION

strongly correlated with the day of the week, with most attacks happening in the middle of the week and relatively few attacks occurring on the weekend. Therefore, a day of the week baseline already gives a moderate Pearson correlation of .52.

The Hackmageddon attack counts have only a .23 autocorrelation for a day-later shift, indicating that the attack counts are not themselves predictive of the next day. The number of documents returned by the webpage query *"news AND cyber AND tech"* for each day provided a strong baseline, with a .55 correlation with Hackmageddon counts. However, a multiple linear regression model combining all event types and top correlating entity counts achieved a correlation of .76, demonstrating that processing the text at a semantic level more accurately captures the magnitude of attacks over time. It should be noted, however, that no single event or entity type in this model had a significantly high coefficient, and only a few event types (*Exploitation* and *Manipulation*) and entity types (hardware and filenames) were significantly correlated on their own within the model. We also tested only event detections or only entity detections, with each achieving a lower score but still above the baseline. This indicates that both event distinctions (lost in the "Entities Only" configuration) and entity distinctions are informative. Results are shown in Table I.

We also experimented using these counts to predict Hackmageddon attack counts one day in the future, with results shown in Table II. We obtained a modest improvement over the baseline of a model that uses only autocorrelation and the day of the week. Attempts to predict Hackmageddon data further out did not beat the baseline method.

One limitation of Hackmageddon for evaluating predictive models is that there is not always a clear causal link between news reports of attacks and future cyberattacks. Some links we do expect are copycat attacks that make use of vulnerabilities that are reported or inferred from reports of attacks, or attacks that signify a larger trend based on some underlying motivation, such as politically-motivated attacks.

Our entity recognition method can capture relevant details of an attack (language, software, etc.) that could be used in future attacks, but we do not model motivations, and so we expect to miss those attacks with our current implementation. Future work incorporating sentiment and behavior models in mini-theories, along with tighter links between events and entitites are areas of future work to address this. By making use of an organization's own private data, however, we could also build a more sophisticated model of attacks and make more informed predictions. The mini-theories framework allows us to incorporate that knowledge tailored for a potential cyberattack victim.

B. Mini-theory Evaluation

To evaluate the viability of the combination of these components for analysis and prediction, we use a simple set of mini-theories to predict cyberattacks against an organization. Our ground truth in this evaluation is private endpoint malware attack data for a single organization. We therefore use extracted event data from webpages, combined with mini-theories, to attempt to predict these reported cyberattacks.

The data used for event and entity detection was collected from the contents of webpages linked from tweets collected from January 2018 through March 2018 for the private data evaluation (roughly 90k documents). Sentence counts per day were collected, as well as entity types and counts for each instance of an entity detected during the day. Thus unique entities that are mentioned more often during a particular day will have more weight in the model.

Based on prior analysis of separate cyberattack data sets (using a prior year of attack data), we built a model with about 20 rules, then pruned by removing low-weight rules during training. We were then left with a small set of rules to encode in Probabilistic Soft Logic, with examples below:

```
1. Tuesday(T) & ValidDay(T) => Cyberattack(T)
2. Infect(S) & Day2(S,T)
       & ValidDay(T) => Cyberattack(T)
3. Deface(S) & Day{1,2}(S,T)
       & ValidDay(T) => Cyberattack(T)
4. Entity(X,S) & RelatedOrg(X) & Infect(S)
        & Day{1,2}(S,T)
       & ValidDay(T) => Cyberattack(T)
5. Vendor(X,S) & VendorImpact(X)
       & Day{1,2}(S,T)
       & ValidDay(T) => Cyberattack(T)
```

The first rule encodes a baseline to represent the fact that cyberattacks tend to happen on Tuesdays, which could be related to "Patch Tuesday", as patches fixing vulnerabilities are released for Microsoft software and software from some other vendors. This opens up a period where unpatched systems can be attacked by hackers who want to exploit the vulnerability before the patch has been applied. Here the variable T represents a specific day, and provided it is a valid day (a necessary predicate for establishing grounding (i.e. assignment of variables) of predicates for inference), cyberattack activity is then predicted on that day. The *DayX* predicate returns a value of 1 if day T is X days later than day S, with a soft falloff of

values for larger intervals. Rules 2 and 3 represent a tendency for reports of malware infections and defacement (e.g., server takedowns, other damage), to precede further attacks one and two days later.

Rules 4 and 5 encode semantic and logical knowledge that would not otherwise fit into a regression framework, with X standing in for an entity, like "Microsoft". Rule 4 finds instances where an extracted entity (with high confidence) is related in some way to the organization, and with a detection of infection events, predicts an attack. Rule 5 was not used in the final model, but serves as another example of incorporating background knowledge into prediction – it finds vendors mentioned frequently on a particular day, and attenuates the warning based on the profile of the vendor – assuming that a cyber-event involving a vendor with widespread software influence could potentially be a larger threat.

We use Max Likelihood MPE (Most Probable Explanation) to learn a weight for each of the above rules, then generate a time-series for cyberattack predictions for each day based on the input data. To evaluate our system, we then perform correlations of the time-series generated by PSL with the time-series ground truth malware detections. We used 2-fold cross-validation on three months of data (January 2018-March 2018). Our best performing model used 11 rules and we averaged a .25 Pearson correlation of the generated time series in the two folds, which did not beat a baseline of only having the Tuesday rule in the model. On the same data set the models were trained on, the Pearson correlation was only .31, which does not seem to indicate overfitting. We also ran multiple linear regression testing a one-day lead time prediction using our events and entity counts as in the prior experiment, but we were unable to achieve significance with that model. This provides some evidence that publicly available text data alone was not applicable for predicting the attacks on this organization over this time frame. However, given the ability for mini-theories to represent causal relationships and unseen events with the integration of additional data, it is possible that text analysis could still play a role in future models that integrate more data, such as applicable internet-facing services or reports from vulnerability database. We, therefore, believe that deeper analysis is needed to generate and train rules that have predictive power.

VII. Discussion

We have demonstrated that natural language processing techniques like event extraction and named entity recognition show promise for large scale text analysis for cyberattack prediction. Event and entity counts can be used as a useful signal in any model for time-series prediction, while more sophisticated, interpretable analysis is possible using our system when better hypotheses about likely threats can be encoded into mini-theories. In future work, we plan to incorporate repository GitHub commits in the repositories of known cybersecurity tools into our mini-theories, encoding hypotheses about software development and vulnerabilities into the cyberattack kill chain model. While we did not achieve predictive power using mini-theories on this private dataset, we believe that with more data of different modalities (such as having access to internet-facing services, detecting port scans, etc), we may be able to develop models that help interpret and predict future cyberattacks. Other areas of future work will be to focus more on high-confidence single instances of described vulnerabilities or attacks, rather than basing our models on larger trends of event or entity mentions.

Further evaluation is needed to verify matching of event details to fully determine what role such a system can play in automatically gathering cyberattack information from the web. However, we believe the accuracy of the event detection, and the ability to test hypotheses about future threats tailored to a particular organization, serves as a strong base for intelligent analysis of such threats when faced with large amounts of data.

References

[1] C. Sauerwein, C. Sillaber, M. M. Huber, A. Mussmann, and R. Breu, "The tweet advantage: An empirical analysis of 0-day vulnerability information shared on twitter," in *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 2018, pp. 201–215.

[2] Q. Le Sceller, E. B. Karbab, M. Debbabi, and F. Iqbal, "Sonar: Automatic detection of cyber security events over the twitter stream," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ser. ARES '17. New York, NY, USA: ACM, 2017, pp. 23:1–23:11. [Online]. Available: http://doi.acm.org/10.1145/3098954.3098992

[3] Lockheed Martin, "Cyber kill chain®," *https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/Gaining_the_Advantage_Cyber_Kill_Chain.pdf*, 2014.

[4] K. Papanikolaou, H. Papageorgiou, N. Papasarantopoulos, T. Stathopoulou, and G. Papastefanatos, "Just the facts with PALOMAR: Detecting protest events in media outlets and Twitter." 2016.

[5] H. Lee, M. Surdeanu, B. MacCartney, and D. Jurafsky, "On the importance of text analysis for stock price prediction," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, 2014.

[6] S. K. Lim, A. O. Muis, W. Lu, and C. H. Ong, "Malwaretextdb: A database for annotated malware articles," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1557–1567. [Online]. Available: http://aclweb.org/anthology/P17-1143

[7] B. J. Dorr, M. Petrovic, J. F. Allen, C. M. Teng, and A. Dalton, "Discovering and characterizing emerging events in big data," in *AAAI Fall Symposium Series*, 2014.

[8] D. Davidson, "Causal relations," *The Journal of Philosophy*, vol. 64, no. 21, pp. 691–703, 1967. [Online]. Available: http://www.jstor.org/stable/2023853

[9] E. Riloff, "Automatically constructing a dictionary for information extraction tasks," in *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 1993.

[10] P. Mannem, C. Ma, X. Fern, P. Tadepalli, T. Dietterich, and J. Doppa, "Oregon state university at tac kbp 2014," in *Proceedings of the NIST TAC KBP 2014 Event Track*, 2014.

[11] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, "Hinge-loss Markov random fields and probabilistic soft logic," *Journal of Machine Learning Research (JMLR)*, vol. 18, no. 109, pp. 1–67, 2017.

[12] N. Blaylock and J. Allen, "Hierarchical goal recognition," in *Plan, Activity and Intent recognition*, G. Sukthankar, C. Geib, H. Bui, D. Pynadath, and R. P. Goldman, Eds. Stroudsburg, PA, USA: Morgan Kaufman, 2014, pp. 3–32.

[13] C. L. Jones, R. A. Bridges, K. M. T. Huffer, and J. R. Goodall, "Towards a relation extraction framework for cyber-security concepts," in *Proceedings of the 10th Annual Cyber and Information Security Research Conference (CISR). ACM International Conference Proceedings Series*, 2015.

[14] P. Passeri, "Hackmageddon," http://www.hackmageddon.com/, 2011, accessed: 2018-01-10.

[15] J. Allen, M. Swift, and W. de Beaumont, "Deep Semantic Analysis of Text," in *Symposium on Semantics in Systems for Text Processing (STEP)*. Morristown, NJ, USA: Association for Computational Linguistics, 2008, pp. 343–354. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1626481.1626508http://speech.ee.ntu.edu.tw/{~}aaron/acl/www.aclweb.org/anthology-new/W/W08/W08-2227.pdf

[16] K. Kipper-Schuler, "VerbNet: A broad-coverage, comprehensive verb lexicon," Ph.D. dissertation, University of Pennsylvania, 2005.

[17] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[18] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, April 2017, pp. 427–431.

[19] M. Honnibal and I. Montani, "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," *To appear*, 2017.

[20] R. A. Bridges, C. L. Jones, M. D. Iannacone, K. M. Testa, and J. R. Goodall, "Automatic labeling for entity extraction in cyber security," in *ASE Third International Conference on Cyber Security, Academy of Science and Engineering (ASE)*, 2014.

[21] N. I. of Standards and T. (U.S.), "National vulnerability database [electronic resource] : automating vulnerability management security measurement and compliance checking," p. :.

[22] A. V. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connect. Sci.*, vol. 7, pp. 123–146, 1995.

[23] J. Pustejovsky, "The role of event-based representations and reasoning in language," in *Proceedings of NAACL Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, 2013.

[24] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, no. 1-2, pp. 107–136, 2006. [Online]. Available: https://homes.cs.washington.edu/{~}pedrod/papers/mlj05.pdfhttp://link.springer.com/article/10.1007/s10994-006-5833-1

APPENDIX

Event	Parent	Definition	ARG0	ARG1	ARG2	ARG3
Reconnaisance (REC)		attacker selects target, researches it, and attempts to identify vulnerabilities.	attacker	target1	resource	
TARGET	REC	selecting an entity for an attack	attacker	target1		
GATHER_RESOURCES	REC	gathering resources e.g. computers	attacker	target1	resource	
GATHER_INFO	REC	gathering info for attack e.g. keys, account numbers	attacker	target1	resource	
PROFILE_TARGET	REC	profiling an established target	attacker	target1		
Weaponize (WEA)		attacker creates remote access malware weapon, such as a virus or worm, tailored to one or more vulnerabilities	attacker	target2	purpose	
MISUSE	WEA	weaponization/exploitation of certain tools or resources towards the attacker's objective	attacker	target2	purpose	
Delivery (DEL)		Intruder transmits weapon to target (e.g., via e-mail attachments, websites or USB drives)	attacker	target1	data	act2
PHISH	DEL	sending deceitful, weaponized communication	attacker	data		
TRICK	DEL	trick or deceit to accomplish delivery (includes directing to a weaponized website)	attacker	person	act2	
Exploitation (EXP)		malware weapon's program code triggers, which takes action on target network to exploit vulnerability.	attacker	target1	vulnerability	
BREAK_IN	EXP	force an entry into a target organization or system to gain data access.	attacker	target1		
EXPLOIT	EXP	exploitation of code vulnerability	attacker	vulnerability	purpose	
Installation (INS)		malware weapon installs access point (e.g., "backdoor") usable by intruder	attacker	target2	malware	
INFECT	INS	affecting a target system with a virus or infection	attacker	target2	malware	
Command and Control (CNC)		malware enables "hands on the keyboard" access to target network, potentially to orchestrate attacks with other devices	attacker	target2	data	purpose
ACCESS	CNC	system or data access and control	attacker	target2	data	purpose
Actions on Objectives (AOO)		attacker takes action to achieve their goals, such as data exfiltration, data destruction, or encryption for ransom	attacker			
ATTACK	AOO	taking an action against a target (e.g., organization, person or system)	attacker	target		
SPY	ATTACK	carry out spying activity	attacker	target		
RENDER_VULNERABLE	AOO	exposing data/vulnerability	attacker	target2	data	
STEAL	REND_VULN	taking data from breached system	attacker	target	data	
DATA_TRANSFER	REND_VULN	leaking, dumping, or selling activity	attacker	data		
MANIPULATE	AOO	other manipulations on the system	attacker	target	theme	
DEFACE	MANIPULATE	cause harm by impairing function or reputation of the target	attacker	target2		
TAKE_DOWN	MANIPULATE	cause system shutdown	attacker	target2		
Speech Act (SA)		communication or info dissemination	agent	information		
NOTIFY	SA	inform affected entities of possible problem (e.g. data breach)	target	customer	information	
REVEAL	SA	divulge information (usually regarding an attack)	informant	information		
INVESTIGATE	SA	carry out an inquiry into a cyber event	informant	act1		
DETECT	SA	discover and identify information (generally, information regarding an attack) post attack	informant	act1		
General Events (GE)		other general relevant events	agent			
DEFEND	GE	resist or deflect an attack	target	act1		
THREATEN	GE	express intent to harm	attacker	act1		
MOTIVATION	GE	motivation behind the aggressor's (attempted) attack	attacker			

TABLE III
THE EVENTS IN THE THREAT ONTOLOGY

Role	Parent	Description	NE Class (1)	NE Class (2)	NE Class (3)
agent		agent of an act	Person	Organization	
attacker	agent	agressor, perpetrator, attacker	Person	Organization	
informant	agent	informant, analyst, media, investigator	Person	Organization	
person	agent		Person		
target		undergoer of an act	Person	Organization	
target1	target	people or organization	Person	Organization	
target2	target	website, db, system, software, server, network	Application	Vendor	URL
act1		cyberattack, offensive attack			
act2		activity target is tricked into doing/visiting	URL		
customer		target's customers	Person		
data		compromised data			
information		information released by informant			
malware		malware, bug	Application	Vendor	
vulnerability		code vulerability	Application	Vendor	URL
resource		resource gathered for an attack	Application	Vendor	Money

TABLE IV
THE ROLES IN THE THREAT ONTOLOGY WITH THE ASSOCIATED NAMED ENTITY (NE) CLASSES